Extended Peer-to-peer Protocol based on IPv6*

† Minji Nam, † Kunwoo Park, † Hosik Cho, † Taekyoung Kwon, † Yanghee Choi, ‡ Taewan You and ‡ Seungyun Lee
† School of Computer Science and Engineering, Seoul National University, Korea
† {mjnam, kwpark, hscho, tkkwon, yhchoi}@mmlab.snu.ac.kr
‡ Electronics and Telecommunications Research Institute, Korea
‡ {twyou, syl}@etri.er.kr

***Abstract* — This paper proposes extended peer-to-peer (P2P) protocol based on IPv6. There are a lot of works on structured P2P protocols using distributed hash table (DHT). Especially, our P2P algorithm and implementation are focused on the efficiency and the scalability for maintaining the P2P overlay network. To achieve efficient routing, we merges P2P overlay with messenger service. With friend nodes as finger, a query can routed through more efficient path on the P2P overlay network. We implemented extended P2P protocol for messenger and multimedia services (MMP2P) on Microsoft Windows environment and evaluated the performance of our proposal.**

***Keywords* — Peer-to-peer, overlay network, distributed hash table, messenger, multimedia.**

1. Introduction

IPv4 is the Internet Protocol version currently in use. IPv4 is in use since the Internet was born and has worked very well until now. However, many experts agree with that IPv6 will be needed in a few years because of the shortage of IPv4 address pool. IPv6 provides a large address space, satisfies the growing need of security experienced by the Internet community and improves mobile network. However, IPv6 is not prevalent yet, merely being adopted in research network. One of the main reasons is the lack of killer application. In this paper, we describe MMP2P (Messenger and Multimedia P2P Protocol) which is an extended Peer-to-Peer protocol base on IPv6. Also, we implemented MMP2P in Microsoft Visual C++ .net which is an advanced programming language for multimedia application built into Microsoft Windows operating system.

2. Related Works

Peer-to-Peer(P2P) technology [1], which is an application level protocol such as HTTP, is defined as a technology for supporting communication, sharing and exchanging resources between users. CHORD [2] and CAN [3] which are representative algorithms of P2P overlay network will be briefly mentioned.

The basic problem of P2P protocol is to efficiently locate the nodes which store the items. CHORD is a distributed lookup protocol that arranges the nodes efficiently. CHORD provides only one operation: matching the nodes with the given key. CHORD can be efficiently reconstructed and can handle the operation while a node joins or leaves frequently.

*This paper was supported in part by the Electronics and Telecommunications Research Institute (ETRI), in part by the Brain Korea 21 project of Ministry of Education, and in part by the National Research Laboratory project of Ministry of Science and Technology, 2004, Korea

The cost for maintaining the system and communicating between nodes increases by each node scaling logarithmically with the number of nodes. Also, it is easy to implement a file sharing program based on CHORD algorithm. File information can be matched with the node by using file name.

Content-Addressable Network (CAN) stores item using virtual Descartes coordinates which is similar to the hash table. The entire space is divided amongst the nodes and each node is allocated its own portion of the space. Data is stored at a point within the space. A point means the individual area each node has. Virtual coordinate space stores (key, value) pairs of a data. A key is deterministically mapped onto a point using a uniform hash function. Also, users put (key, value) pair into hash function to find the area of a data and to search the data within the area.

3. Extended P2P Protocol

We suggest P2P overlay protocol based on IPv6 which has several benefits compared to the one based on IPv4. When a new node first joins the overlay network, it sends join request message to the key node. It is assumed that the key node is the special node which is always connected and every node knows the address of it. A new node which joins overlay network successfully is allocated a resource locator. A resource locator involves addresses of left and right neighbor node as well as addresses of friends which are used in instance messenger. When a user requests to find a data, this request message is sent to a node whose ID is proximity to resource locator of the data, and the node replies with the resource locator of the data. Then, the user obtains the real address of the data. In MMP2P, we use friends list as a short cut reference to send a request message to the node which has the data on overlay network.

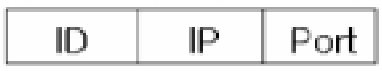

Figure 1. Resource Locator Format. Identifiers are generated by hashing e-mail address, file name, or resource name. Resource locator contains idintifer and the location of that identifier, that is IPv6 address and port number.

Overlay nodes use Distributed Hash Table (DHT). Individual node maintains DHT which stores resource locators of data. If a node receives a request message of the data which it has, it replies with the resource locator containing information about real location.

A resource locator contains data access information: IP address and port. Figure 1 shows the resource locator format. Resource is distinguished with ID. ID is either a User ID or file ID. User ID means a user E-mail address and file ID means the result from MD5 hashing the file name [4].

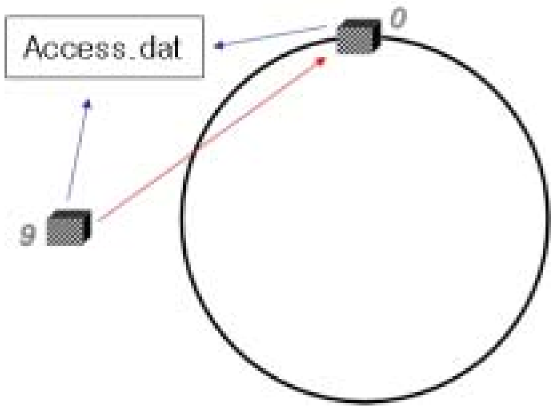

Figure 2. Bootstrap operation. A node want to join to existing P2P overlay network should know at least one node already participating to the overlay. Information about where to try to connect is stored in access.dat and the information changes when it successfully join to the P2P overlay network.

At the beginning, a new node learns about its appropriate location and neighbors from key node. The information of key node is in access.dat file which is generated when a user runs the program. Access.dat contains key node's IP and port information and key node's ID is always zero.

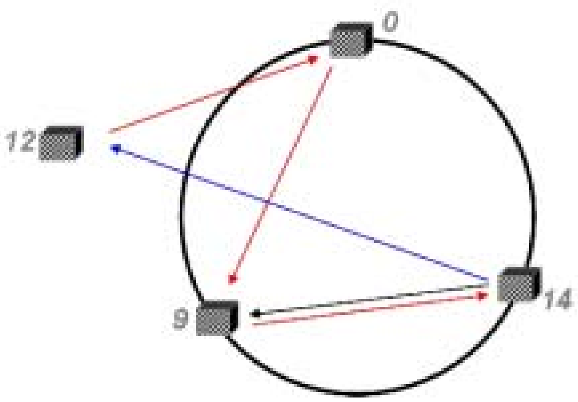

Figure 3. Node 12 joins into the overlay network. Node 12 send JOIN message to the key node, and that quary is routed through overlay network. Node 14 has responsiblity for joining the node 12, and send neighbour update message to the node 12 and node 9.

Figure 3 shows the process that a new node joins existing overlay network. Node 0, node 9, node 14 are forming an overlay network. When node 12 wants to join the overlay network, it sends JOIN message to node 0 which is a key node. The key node relays it to node 14 which is in charge of handling the request. Then, node 14 replies to node 12 information of its left and right neighbor. Also, node 14 informs node 9 that the right neighbor has changed from node 14 to node 12 and node 14 updates itself that its left neighbor is node 12. By above process, node 12 can join into the existing overlay network successfully.

Nodes can join and leave at any time without any notification. In order to preserve overlay network, every node has four resource locations: its left and right neighbor nodes and their neighbor nodes. Also, it has resource locators of friends to maintain the friends list.

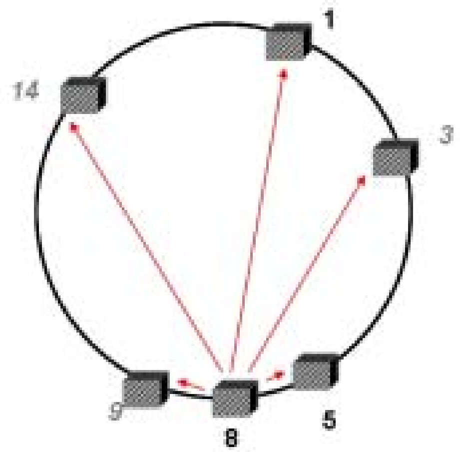

Figure 4. Routing Logic. Node 8 maintains fingers toward its friends and neighbors. When node 8 creates or receives a query for a resource, the node decides which friend or neighbor is closest to that requested resource.

In figure 4, node 8 has friends list: node 1, node 3 and node 14. Its neighbors are node 5 and node 8. If node 8 sends a file search message, whose file ID is 16, this message will not be relayed from node 9 to node 14 but will be relayed directly to node 14. In that, MMP2P uses friends list as a short cut reference. Even if the overlay network size is large enough, users can find data within several hop relays.

On overlay network, a resource locator is changed whenever a node joins or leaves the network. Each node should maintain the resource locator to adapt changes. So nodes send a STORE message to other nodes periodically. A STORE message contains every file ID in the shared directory of itself. A STORE message is relayed around the overlay network, compared with each node ID and stored in appropriate location. If the time of a STORE message of a data is expired before new one arrives, then the node regards as the data has been deleted and removes the resource locator of the data.

4. Performance Evaluation

Let N be the number of users and M be the average number of friends which each user has in the friends list. In order to find the location of a file, only O(N/M) nodes relay the

SEARCH message. Also, if there are total T files on the overlay network then each node needs to maintain average T/N resource locators. Since IDs are spread across the overlay network nodes using a uniform hash function, the number of IDs each node has to maintain is indicative of the average number of the data the node stores.

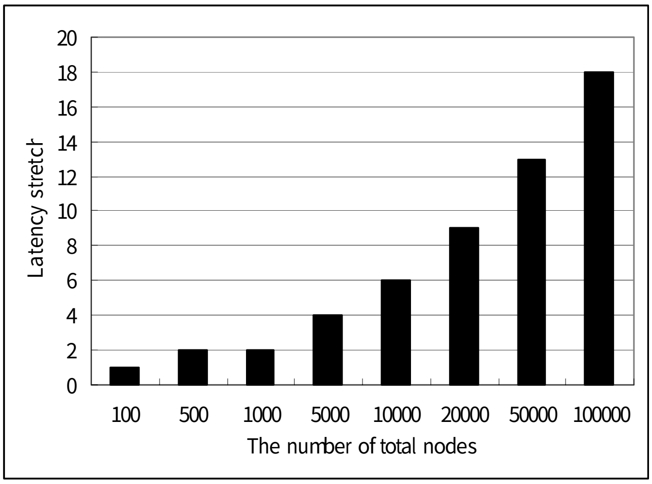

Figure 5. Latency stretches while friend node is pegged at 50.

We expected each instant messenger users including MSN, ICQ, AOL, etc., keeps approximately 50 or more friends in their friend list. Therefore, we have simulated Figure 5, latency stretch versus increasing number of total nodes with 50 friend list short cuts. Although constant size of storage is needed, we can get nearly log scale latency stretch behavior.

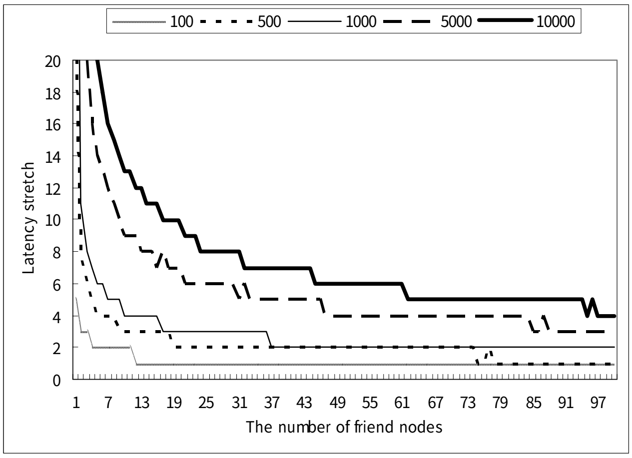

Figure 6. The numbers of friend nodes have an effect on latency stretch. The total number of nodes in the P2P overlay network increases from 100 to 10000.

Friend nodes act very important role in MMP2P algorithm. They are the essence of MMP2P and the key of log scale performance using only constant storage space. Figure 6 shows the latency stretch while friend nodes vary from 1 to 100.

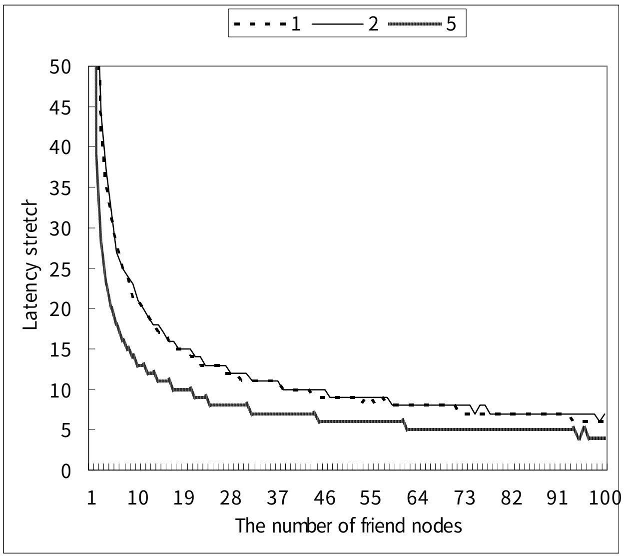

Figure 7. Latency stretches versus number of friend nodes for different number of backup branches.

Every node maintains typical number of backup branches. Usually each node has first step backup branch to support overlay network and second step backup branches for node failure recovery process. However, to improve the performance of latency stretch, we can set the backup branches up to five. Figure 7 shows the impact of backup branches contribute to performance.

CHORD like protocols maintain a number of fingers in finger table. However, the size of finger table should be increased as the number of total nodes constituting the overlay network increases. Furthermore, CHORD like approach needs finger updating cost as nodes join or leave dynamically. In comparison with CHORD like approach, our protocol maintains the fixed maximum number of friends' information and does not need any additional maintenance overhead when the topology changed. Mean latency stretch is also slightly less than CHORD like approach. So we can say that our extended P2P protocol provides certain level of scalability and efficiency [5][6].

5. Services Using MMP2P

Our overlay P2P algorithm uses the information of friends to shorten the routing path. As our protocol is tightly coupled with instant messenger service, we also implemented some applications that work on extended P2P protocol. Basically, instant messenger service is available, which uses e-mail address as identifier. Figure

Figure 8. Messenger service of MMP2P. Group create and delete, friend associate and remove, and current state chage are available. As e-mail addresses are used as identifier, MMP2P provides security using password when duplicated identifier detected.

P2P overlay algorithms are designed for efficient resource management and searching. Resource locators for resources are scattered on the whole overlay network. So our protocol also provides file sharing among friends and file searching over the whole P2P overlay participants.

Finally, we also suggested overlay multicast algorithm employs P2P overlay information [7][8]. Using overlay multicast, we implemented group conference tool and video on demand (VoD) service as the integrated services of P2P overlay instant messenger.

Figure 9. File sharing and searching service of MMP2P. A user can browse registered friends' file list and search file name on the whole overlay P2P network.

Figure 10. Overlay multicast service of MMP2P. To initiate conference with friends, select friends want to gathering and press the multicast button. Then selected friends receive multicast join request and construct overlay multicast tree. Using that multicast tree, group conferencing, multiple file transmit, and video on demand services are available.

6. Conclusion

We suggest MMP2P which is an extended P2P protocol based on IPv6 and we have implemented several applications for Windows using MMP2P: instant messenger, file sharing function and high-quality multimedia conference system. Development environments were Microsoft Visual C++ .net for Window XP. Our results show that MMP2P improves the response time compared to CHORD and CAN. Also, MMP2P is easy to implement. MMP2P can be expected to contribute in IPv6 deployment offering foundation of Peer-to-Peer efficiency communication between users.

REFERENCES

[1] David Barkai, "Technologies for Sharing and Collaborating on the Net," Proceedings of the 1st International Conference on Peer-to-Peer Computing, August 2001.

[2] Stoica, I. Morris, R. Liben-Nowell, D. Karger, D.R. Kaashoek, M.F. Dabek, F. Balakrishnan, H. "Chord: a scalable peer-to-peer lookup protocol for Internet applications", Networking, IEEE/ACM Transactions on , Volume: 11 , Issue: 1 , Pages:17 - 32, Feb. 2003

[3] Ratnasamy, Paul Francis, Mark Handley, Richard Karp and Scott Shenker, "A Scalable Content-Addressable Network", SIGCOMM, 2001

[4] RFC 1321, "The MD5 Message-Digest Algorithm"

[5] Romgmei Zhang, Y. Charlie Hu, and Peter Druschel, "Optimizaing Routing in Structured Peer-to-Peer Overlay Networks Using Routing Table Redundancy," Proceeding of the 9th IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS'03).

[6] Daniel A. Menasce, "Scalable P2P Search," IEEE Internet Computing, March-April 2003.

[7] Tetsuya Oh-ishi, Koji Sakai, Kazuhiro Kikuma, and Akira Kurokawa, "Study of the Relationship between Peer-to-Peer System and IP Multicasting," World Telecommunications Congress 2002 (WTC2002), January 2003.

[8] Adam Weirzbicki, Robert Szczepaniak, and Marcin Buszka, "Application Layer Multicast For Efficient Peer-to-Peer Applications." Proceedings of the 3rd IEEE Workshop on Internet Applications (WIAPP'03).